REMARKS

Claims 21-22 and 24-36 are pending in this application, of which claims 21 and 27 are independent. Claim 23 has been canceled. Claims 1-20 were previously canceled. Applicants acknowledge the Examiner's withdrawal of the previous objections to the specification.

In an effort to expedite prosecution, Applicants have amended the claims to place the claims in a more suitable U.S. format. No new matter has been added.

Allowance of claims 21-22 and 24-36 is respectfully requested.

Objection to the Specification

Applicants have amended the Abstract. No new matter has been added.

Drawing Objection

The drawings were objected to because they allegedly fail to comply with 37 CFR § 1.84(o). The Action asserts that this non-compliance arises because "Figures 1-2 do not contain suitable legends or object labels that adequately indicate what the drawings consist of without referring in detail to the specification." Action at page 2, paragraph 2. Applicants respectfully traverse this objection.

The cited CFR section does not include a requirement that the specification should not be referred to when determining what the drawings consist of. Applicants conducted a thorough search of the MPEP, CFR and USC to find the source of the alleged requirement and respectfully submit that support for such a requirement does not exist. Accordingly, Applicants request withdrawal of this objection.

The objection to the drawings is overly vague and therefore constitutes a failure by the Examiner to expeditiously provide information necessary to resolve issues. Such a vague objection needlessly encourages piecemeal prosecution, which is to be avoided. See, MPEP § 707.07(g). Accordingly, in the event that the Office maintains the drawing objection, Applicants request that the Office cite specific MPEP, CFR or USC sections that support the objection.

Rejections Under 35 U.S.C. § 112

Applicants submit that the claim amendments render these rejections moot, and request that they be withdrawn.

Rejections Under 35 U.S.C. § 103

Claims 21-36 stand rejected under 35 U.S.C. 103(a) as unpatentable over U.S. Patent No. 6,363,431 (Barker), U.S. Patent No. 6,349,333 (Panikatt) and U.S. Patent Pub. No. 2002/0016867 (Kampe). Applicants respectfully traverse this rejection. Independent claims 21 and 27 recite features that are not disclosed in the documents relied upon by the Examiner, regardless of whether these documents are considered individually or in the combination asserted by the Examiner. For example, the Action points to nothing in the cited references that discloses applicants' claim 21 feature of logging possible events in a client event service for initializing or updating the client.

Exemplary embodiments encompassed by Applicants' claims are directed to a method and system for managing and transmitting events from a server to a client in which the client sees data transmission being initiated by the server. In an exemplary method, for every event to be transmitted from a server via a communication link to a client (for example, a client application), the event is logged using a client event service and a server event service. Events for which logging has been performed are transmitted from the server to the client. Such event logging prompts a respective update or, for example, a first logging can prompt an initialization of the client/server system.

When an event occurs, it is first reported to an installation interface of the server. If the event in question has been logged, it is transferred from the installation interface to the server event service. The client event service uses the communication link to make requests for event transmission to the server event service. If there is an event that has been detected by the server event service, it is transmitted via the communication link to the client event service based on the received request.

Within the client, the client event service transmits received events to the client application, where the event is reported, for example, by producing an entry that describes the event in an event list. Transmitting an event that has occurred to the client application can therefore avoid no active requests from the client application. Because the client application does not communicate with the server, but rather with the client event service, it can be independent of the server. When the method is used, the client application sees the event handling operation taking place as in a local environment.

In an exemplary embodiment, the client application logs an appropriate client callback function in the client event service for an event about which it is to be notified. The client event service then uses the communication link to log a corresponding server callback function in the server event service. This logging is carried out separately for an event about which the client application is to be notified. In this way, it is possible to handle events independently of one another.

An association can be made (for example, in a list) in preparation for the method, so that this association can be used to assign a unique name to events possibly occurring in the installation. This association exists in the client and in the server. Thus, an event can have the same associated name in the client and in the server. To log the callback functions, the client application calls a client logging function from the client event service and provides it with a name of an event in question and with a pointer to the client callback function that is to be logged.

The client logging function can then generate a unique event identifier and transmits this event identifier together with the event name to a server logging function of the server event service via the communication link.

The server logging function logs a server callback function with the installation interface by transferring the event name. The server logging function can then store a data record, which contains the event identifier, a pointer to the server callback function which is to be logged and possibly further data, such as the event name, in a server event table. The server logging function can then use the communication link to report back to the client logging function of the client event service that the logging operation has been performed. The client logging function then logs the client callback function by storing a data record, which contains the event identifier, a pointer to the client callback function that is to be logged and possibly further data, such as the event name, in a client event table.

Applicants respectfully submit that such features are neither disclosed nor suggested by Barker, Panikatt and Kampe, viewed alone or in combination. For example, the Action points to nothing in the cited references that discloses Applicants' claim 21 feature of logging possible events in a client event service for initializing or updating the client.

The cited sections in Barker (col. 4, lines 19-36 and col. 38, line 50-col. 39, line 16) disclose that client commands generate HTTP requests to the element management system server. The server gathers the information, dynamically generates a web page, and sends results/outputs to a web browser for display.

Barker discloses Java applets which, although referred to by the Examiner as equivalent to the claimed "client event service", are in fact client applications. See Barker at col. 4, lines 27-30. In the Examiner's rejection, the Java applets therefore must function as <u>both</u> client applications and as a client event service. However, Applicants' claims recite an architecture and functionality which include client applications that are separate and distinct from a client event server. As Applicants note on page 3, lines 13-21 of their specification, the client applications do not communicate with the server but rather communicate with the client event service which is independent of the server.

Thus, Barker does not disclose a client event service as presently claimed. In addition, the Action does not specify which features in Barker perform the claimed logging function of the client event server. For example, the Barker patent fails to disclose the presently claimed feature of logging possible events in a client event service for initializing or updating the client.

Regarding the claimed logging of possible events in a server event service for initializing or updating the server, the Action equates Applicants' server event service with "various EMS server subcomponents" and refers to Barker's Fig. 4, which contains a multitude of components, as well as Barker at col. 4, lines 37-55. However, this rejection is overly vague, and again constitutes a failure by the Office to expeditiously provide information necessary to resolve issues. Accordingly, in the event that the Office maintains this rejection, Applicants request that the Office cite specific components in Barker that support the rejection.

Barker is cited as disclosing at col. 11, lines 21-28 Applicants' claimed sending of requests initiated by the client event service regarding the detected events to the server event service. Applicants again disagree that Barker provides any disclosure of such a feature. The cited section in Barker states that clients register a filter with an Event Distributor of a server to request delivery (via a callback function) of events matching the filter. However, Barker's registering a filter, even under a broadest reasonable interpretation, does not constitute sending of requests initiated by a client event service as presently claimed. Moreover, Applicants' client application does not communicate with a server, but rather with the presently claimed client event service (which is independent of a server).

Barker's definition of client callback function in Fig. 6 is cited as disclosing the Applicants' claimed feature whereby a client application logs a client callback function in a client event service for every event about which it is to be notified. However, Barker's client callback function is disclosed as merely a function that is passed from the client to the server. Thus, there is no support in Barker for a client application that logs a client callback function in a client event service for every event about which it is to be notified.

The logging of callback functions for an event with which a same event name is associated with the client and with the server, as presently claimed, is also lacking in the documents relied upon by the Examiner for at least reasons similar to those already discussed. For example, in contrast to the method and system of Barker, Applicants' client application does not communicate with a server, but rather only with the client event service.

The Panikatt and Kampe patents fail to overcome the deficiencies of the Barker patent. For example, Panikatt is cited as allegedly teaching a callback function as a server callback function, and Kampe is cited as teaching a client event table which contains a client callback pointer. Such features do not correspond to features as presently claimed. Moreover, these patents do not disclose the features already discussed, even when considered in combination with the Barker patent.

As such, Applicants' claim 21 is allowable. Claim 27 recites similar features, such as a client event service, and is separately allowable.

All of the remaining claims depend from claims 21 or 27, and add further distinguishing features. As such, these claims are also allowable.

Conclusion

For the foregoing reasons, Applicants respectfully submit that this application is in immediate condition for allowance and all pending claims are patentably distinct from the cited references. Reconsideration and allowance of all pending claims are respectfully requested.

In the event that there are any questions about this application, the Examiner is requested to telephone Applicants' undersigned representative so that prosecution of the application may be expedited.

If additional fees are required for any reason, please charge Deposit Account No. 02-4800 the necessary amount.

Respectfully submitted,

BUCHANAN INGERSOLL & ROONEY PC

Date: May 11, 2009

By: Peter C Keane Reg No 32,858
for Brian N. Fletcher
Registration No. 51683

P.O. Box 1404
Alexandria, VA 22313-1404
703 836 6620